

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA: CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP**  
**LẬP TRÌNH VỚI PYTHON**  
**“Xây dựng Website bán hàng thời trang trực tuyến”**

**Giảng viên hướng dẫn** : Ninh Thị Thu Trang

**Nhóm lớp** : 01

**Nhóm BTL** : 10

**Sinh viên thực hiện** : Bùi Nguyên Giáp – B22DCAT101  
Trần Đình Lương – B20DCCN414  
Đinh Thị Thanh Tâm – B22DCAT253

**HÀ NỘI, THÁNG 5 NĂM 2025**

# MỤC LỤC

MỤC LỤC.....	2
DANH MỤC CÁC HÌNH VẼ.....	4
DANH MỤC CÁC BẢNG BIỂU .....	4
PHÂN CÔNG NHIỆM VỤ NHÓM THỰC HIỆN .....	5
lời mở đầu.....	6
<b>CHƯƠNG 1. Tổng quan</b> .....	7
1.1 Lý do chọn đề tài .....	7
1.2 Mục tiêu.....	7
1.3 Công nghệ sử dụng.....	7
1.3.1 Ngôn ngữ lập trình .....	7
1.3.2 Framework / Kiến trúc .....	7
1.3.3 IDE (môi trường phát triển) .....	8
1.3.4 Hệ quản trị cơ sở dữ liệu .....	8
1.3.5 Giao diện người dùng (Front-end) .....	8
1.4 Kết chương .....	8
<b>CHƯƠNG 2. Phân tích hệ thống</b> .....	9
2.1 Xác định yêu cầu. ....	9
2.1.1 Mục đích phạm vi, phạm vi hệ thống.....	9
2.1.2 Chức năng hệ thống: .....	9
2.2 Phân tích cơ sở dữ liệu .....	9
2.2.1 users (Khách hàng).....	9
2.2.2 products (Sản phẩm).....	10
2.2.3 Orders (Đơn hàng).....	10
2.2.4 Category (danh mục).....	12
2.2.5 payment (thanh toán).....	12
2.3 Mối quan hệ giữa các đối tượng trong cơ sở dữ liệu.....	13
2.3.1 Category và Product: .....	13
2.3.2 User và Order: .....	13
2.3.3 Order và OrderItem: .....	13
2.3.4 Product và OrderItem: .....	14
2.4 Thiết kế cơ sở dữ liệu .....	14
2.5 Thiết kế giao diện .....	14
2.6 Kết chương .....	18

<b>CHƯƠNG 3. Mô tả cách hoạt động của hệ thống .....</b>	<b>19</b>
3.1 Các thành phần chính .....	19
3.2 Cấu trúc thư mục .....	19
3.3 Quy trình hoạt động tổng quát.....	20
<b>3.3.1</b> Tổng quan hoạt động hệ thống.....	20
<b>3.3.2</b> Tương tác nội bộ giữa các thành phần .....	21
3.4 Kết chương .....	21
Kết luận, kiến nghị .....	22
Tài liệu tham khảo.....	23

## DANH MỤC CÁC HÌNH VẼ

Hình 1 Lớp CreateUserForm, xử lý việc đăng ký tài khoản người dùng mới .....	10
Hình 2 Mô hình dữ liệu Products .....	10
Hình 3 Mô hình dữ liệu Order.....	11
Hình 4 Mô hình dữ liệu OrderItem .....	11
Hình 5 Mô hình dữ liệu Category .....	12
Hình 6 Mô hình dữ liệu Payment_VNPay và biểu mẫu PaymentForm .....	12
Hình 7 Thiết kế cơ sở dữ liệu .....	14
Hình 8 Giao diện trang đăng nhập của website bán hàng .....	15
Hình 9 Giao diện trang chủ sau khi người dùng đăng nhập của website .....	15
Hình 10 Giao diện đăng ký tài khoản của website .....	16
Hình 11 Giao diện giỏ hàng (cart) của người dùng.....	16
Hình 12 Giao diện trang thông tin cá nhân (profile) của người dùng đã đăng nhập.....	17
Hình 13 Giao diện danh sách sản phẩm theo danh mục.....	17
Hình 14 Giao diện người quản trị.....	18
Hình 15 Giao diện Dashboard – Quản lý danh mục sản phẩm .....	18
Hình 16 Cấu trúc thư mục .....	19

## DANH MỤC CÁC BẢNG BIỂU

Bảng 1. Bảng mô tả tương tác nội bộ giữa các thành phần.....	21
--	----

## PHÂN CÔNG NHIỆM VỤ NHÓM THỰC HIỆN

	<b>Trần Đình Lương</b>	<b>Bùi Nguyên Giáp</b>	<b>Đinh Thị Thanh Tâm</b>
<b>Phân tích/ Thiết kế</b>	BA dự án, phân tích thiết kế database	Hỗ trợ thiết kế, đóng góp ý kiến xây dựng dự án	Hỗ trợ thiết kế, đóng góp ý kiến xây dựng dự án
<b>Front end</b>	Xây dựng trang chủ, lọc sản phẩm theo thể loại, thêm sản phẩm vào giỏ hàng, thanh toán	Xây dựng trang quản trị: nắm bắt thông tin user, sản phẩm, danh mục	Xây dựng đăng nhập, đăng kí, update thông tin, tài khoản, phân quyền
<b>Back end</b>	Xử lí Back end cho giỏ hàng, order sản phẩm, danh mục khách hàng, thanh toán online-vnpay	Xử lí back end cho module quản trị (user, sản phẩm, danh mục)	Xử lí back end cho module người dùng (đăng nhập, đăng kí, update thông tin, phân quyền)
<b>Chung</b>	Xây dựng Front end website	Xây dựng Front end website	Xây dựng Front end website

## LỜI MỞ ĐẦU

Trong thời đại công nghệ thông tin đang phát triển mạnh mẽ và thu hút sự quan tâm từ nhiều độ tuổi khác nhau, website đã trở thành một phương tiện linh hoạt và phong phú với đa dạng nội dung, từ tin tức trong nước và quốc tế đến các trang chủ của trò chơi, các nền tảng học trực tuyến, và vô vàn tiện ích khác mà người dùng có thể dễ dàng tiếp cận. Trong số đó, các website bán hàng trực tuyến đang ngày càng được ưa chuộng nhờ vào sự tiện lợi và đa dạng sản phẩm mà chúng mang lại. Khách hàng có thể mua sắm các mặt hàng yêu thích từ bất cứ nơi đâu, bất kỳ thời điểm nào chỉ với một vài thao tác trên thiết bị có kết nối mạng, mà không cần phải đến tận cửa hàng.

Nắm bắt xu thế này, nhóm chúng em đã quyết định thử sức và phát triển bản thân với mong muốn góp phần vào sự phát triển của ngành thương mại điện tử tại Việt Nam. Do đó, nhóm đã chọn đề tài “Nghiên cứu và phát triển Website bán hàng thời trang trực tuyến” nhằm tạo ra một nền tảng tiện lợi và hiệu quả cho người dùng.

Báo cáo bao gồm 5 nội dung chính sau:

- Chương 1: Tổng quan
- Chương 2: Phân tích hệ thống
- Chương 3: Mô tả cách thức hoạt động
- Kết luận và kiến nghị
- Tài liệu tham khảo

# CHƯƠNG 1. TỔNG QUAN

## 1.1 Lý do chọn đề tài

Thời đại công nghệ ngày càng phát triển, nhu cầu mua sắm trực tuyến đã tăng vọt và trở thành một phần không thể thiếu trong cuộc sống hiện đại. Sự phát triển của công nghệ thông tin đã giúp việc mua sắm trực tuyến trở nên dễ dàng và tiện lợi hơn bao giờ hết. Người tiêu dùng có thể tìm kiếm và mua hàng từ bất kỳ đâu, bất kỳ lúc nào, với các tiện ích và lợi ích nổi bật như:

- Tiết kiệm thời gian và chi phí: Khách hàng không cần phải di chuyển đến cửa hàng, giúp tiết kiệm thời gian và chi phí đi lại.
- Linh hoạt về thời gian và địa điểm mua sắm: Khách hàng có thể mua sắm 24/7, phù hợp với lịch trình cá nhân.
- Tiếp cận đa dạng sản phẩm và thương hiệu: Người dùng có thể dễ dàng tìm kiếm nhiều sản phẩm khác nhau từ nhiều thương hiệu trong cùng một nền tảng.
- Tích hợp nhiều hình thức thanh toán và giao hàng tiện lợi: Mua sắm trực tuyến cho phép khách hàng chọn phương thức thanh toán và giao hàng linh hoạt, từ thanh toán trực tiếp đến giao hàng tận nơi.

Với những lợi ích trên, thương mại điện tử đã trở thành xu hướng mua sắm của hiện tại và tương lai. Nắm bắt được nhu cầu này, nhóm chúng em quyết định chọn đề tài: “Xây dựng website bán hàng thời trang trực tuyến.”

## 1.2 Mục tiêu

Phát triển một website bán hàng thân thiện với người dùng, cho phép người dùng dễ dàng tìm kiếm sản phẩm theo từ khóa, danh mục mà họ muốn. Người dùng có thể xem thông tin chi tiết về sản phẩm, đọc đánh giá từ khách hàng khác, thêm sản phẩm vào giỏ hàng và thực hiện các bước thanh toán. Ngoài ra, người dùng còn có thể tìm hiểu thêm thông tin từ các bài viết tư vấn mua sắm và đánh giá sản phẩm trên website.

Đối với người quản lý, website cung cấp một giao diện quản lý trực quan và hiệu quả. Họ có thể thống kê lượt truy cập, quản lý số lượng đơn hàng, theo dõi doanh thu theo từng sản phẩm và đưa ra các quyết định kinh doanh dựa trên dữ liệu thực tế.

Việc phát triển website này không chỉ giúp giải quyết các khó khăn trong việc mua sắm của khách hàng mà còn mang lại nhiều lợi ích khác như tiết kiệm thời gian, nâng cao trải nghiệm mua sắm và góp phần thúc đẩy sự phát triển của thương mại điện tử.

## 1.3 Công nghệ sử dụng

### 1.3.1 Ngôn ngữ lập trình

- Python: Ngôn ngữ chính được sử dụng để xây dựng toàn bộ logic xử lý phía máy chủ. Đây là ngôn ngữ mạnh mẽ, dễ đọc, và được hỗ trợ bởi nhiều thư viện phục vụ lập trình web.

### 1.3.2 Framework / Kiến trúc

Django là một web framework Python theo mô hình MVT (Model - View - Template) – một biến thể của kiến trúc MVC (Model - View - Controller) truyền thống, giúp phát triển web nhanh chóng và an toàn

Các thành phần tiêu biểu:

- `models.py`: Quản lý cấu trúc và thao tác cơ sở dữ liệu (Model).
- `views.py`: Xử lý logic và phản hồi các yêu cầu từ người dùng (Controller).
- Các tệp HTML trong thư mục `templates/`: Giao diện hiển thị (View).

### ***1.3.3 IDE (môi trường phát triển)***

**Visual Studio Code (VS Code):**

- Trình soạn thảo mã nguồn được sử dụng để phát triển và quản lý toàn bộ dự án.
- Hỗ trợ nhiều tiện ích mở rộng phù hợp với lập trình Django.

### ***1.3.4 Hệ quản trị cơ sở dữ liệu***

**SQLite:**

- Là hệ quản trị cơ sở dữ liệu quan hệ nhẹ, được tích hợp sẵn trong Django và phù hợp cho giai đoạn phát triển.
- File dữ liệu: `db.sqlite3`.

### ***1.3.5 Giao diện người dùng (Front-end)***

**Ngôn ngữ và công nghệ giao diện:**

- HTML, CSS, JavaScript: Được sử dụng để xây dựng các trang giao diện người dùng.

**Thư viện và framework hỗ trợ:**

- Bootstrap: Hỗ trợ tạo giao diện hiện đại, responsive, dễ sử dụng.
- jQuery: Thư viện JavaScript hỗ trợ thao tác với DOM, xử lý sự kiện và AJAX dễ dàng hơn.

## **1.4 Kết chương**

Chương 1 đã trình bày khái quát về lý do chọn đề tài, mục tiêu xây dựng hệ thống, cũng như các công nghệ chủ đạo được sử dụng. Việc lựa chọn Django cùng các công cụ hỗ trợ như SQLite, Bootstrap giúp nhóm xây dựng hệ thống hiệu quả, dễ mở rộng và phù hợp với bài toán thực tế trong thương mại điện tử.



## CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG

### 2.1 Xác định yêu cầu.

#### 2.1.1 Mục đích phạm vi, phạm vi hệ thống.

Website cung cấp một nền tảng mua sắm trực tuyến linh hoạt và hiệu quả cho người dùng thể hiện sự thay đổi mang tính đổi mới trong lĩnh vực công nghệ thông tin:

- Kiểu phần mềm: Website
- Người dùng:
  - Người mua: sử dụng để mua sắm, phục vụ nhu cầu
  - Người bán: sử dụng để cung cấp sản phẩm cho khách hàng

#### 2.1.2 Chức năng hệ thống:

##### 1. Dành cho người mua hàng:

- Đăng ký/đăng nhập tài khoản
- Xem danh sách sản phẩm theo danh mục
- Tìm kiếm sản phẩm theo tên, giá, thương hiệu...
- Xem chi tiết sản phẩm (hình ảnh, mô tả, giá...)
- Thêm sản phẩm vào giỏ hàng
- Đặt hàng và thanh toán
- Cập nhật thông tin cá nhân

##### 2. Dành cho admin:

- Quản lý sản phẩm (thêm, sửa, xóa)
- Quản lý user
- Quản lý danh mục sản phẩm
- Quản lý danh sách order

### 2.2 Phân tích cơ sở dữ liệu

Mô hình đại diện cho các thực thể được sử dụng trong trang website: Users, Products, Orders, Cart. Dưới đây là phân tích chi tiết từng mô hình:

#### 2.2.1 users (Khách hàng)

```
class CreateUserForm(UserCreationForm):  
    class Meta:  
        model = User  
        fields = ['username', 'email', 'first_name', 'last_name', 'password1', 'password2']  
  
User = get_user_model()
```

Hình 1 Lớp CreateUserForm, xử lý việc đăng ký tài khoản người dùng mới

**Trường dữ liệu (Fields):**

- username: Một mối quan hệ one-to-one với mô hình User của Django
- email: Trường lưu email người dùng
- first\_name: Trường lưu tên người dùng
- last\_name: Trường lưu họ người dùng
- password: Trường lưu mật khẩu tài khoản

### 2.2.2 products (Sản phẩm)

```
class Product(models.Model):
    category = models.ManyToManyField(Category, related_name='product')
    name = models.CharField(max_length=255, null=True)
    price = models.FloatField(default=0, null=True, blank=False)
    image = models.ImageField(null=True, blank=True)
    detail = models.TextField(null=True, blank=True)

    def __str__(self):
        return self.name if self.name else "Unnamed Product"

    @property
    def imageURL(self):
        try:
            url = self.image.url
        except:
            url = ''
        return url
```

Hình 2 Mô hình dữ liệu Products

**Trường dữ liệu:**

- name: Trường ký tự (tối đa 255 ký tự) để lưu tên sản phẩm.
- image: Trường ảnh để tải lên ảnh sản phẩm (tùy chọn).
- price: Trường số nguyên dương để lưu giá sản phẩm.
- detail: Trường để mô tả sản phẩm.
- category: Trường lưu danh mục sản phẩm

**Phương thức:**

- Self.name(): Trả về tên của sản phẩm khi in đối tượng Product.
- Self.image.url: Trả về hình ảnh

### 2.2.3 Orders (Đơn hàng)

```

class Order(models.Model):
    customer = models.ForeignKey(User, on_delete=models.SET_NULL, null=True, blank=True)
    date_order = models.DateTimeField(auto_now_add=True)
    complete = models.BooleanField(default=False, null=True, blank=False)
    transaction_id = models.CharField(max_length=255, null=True)

    def __str__(self):
        return str(self.customer)

    # Tính tổng item
    @property
    def get_cart_items(self):
        orderitems = self.orderitem_set.all()
        total = sum([item.quantity for item in orderitems])
        return total

    # Tính tổng tiền
    @property
    def get_cart_total(self):
        orderitems = self.orderitem_set.all()
        total = sum([item.get_total for item in orderitems])
        return total

```

*Hình 3 Mô hình dữ liệu Order*

```

class OrderItem(models.Model):
    product = models.ForeignKey(Product, on_delete=models.SET_NULL, null=True, blank=True)
    order = models.ForeignKey(Order, on_delete=models.SET_NULL, null=True, blank=True)
    quantity = models.IntegerField(default=0, null=True, blank=False)
    date_added = models.DateTimeField(auto_now_add=True)

    # Xử lý total
    @property
    def get_total(self):
        return self.product.price * self.quantity

    def __str__(self):
        return str(self.order)

```

*Hình 4 Mô hình dữ liệu OrderItem*

### **Trường dữ liệu:**

- customer: Khóa ngoại đến mô hình User, có thể null. Đại diện cho khách hàng đặt hàng.
- product: Khóa ngoại đến mô hình Product, có thể null. Liên kết đơn hàng với sản phẩm.
- email: Trường lưu email khách hàng.
- address: Trường lưu địa chỉ giao hàng.
- quantity: Trường số lượng mặt hàng.
- Date\_added: Trường ngày tự động thêm giá trị hiện tại khi tạo đơn hàng.
- complete: Trường lưu đơn hàng đã giao thành công chưa

- transaction\_id: Trường lưu ID giao dịch

**Phương thức:**

- get\_cart\_items(): Tính tổng số mặt hàng.
- get\_cart\_total: Tính tổng tiền hàng.

#### 2.2.4 Category (danh mục)

```
class Category(models.Model):
    sub_category = models.ForeignKey('self', on_delete=models.CASCADE, related_name='sub_categories', null=True,
                                     blank=True)
    is_sub = models.BooleanField(default=False)
    name = models.CharField(max_length=255, null=True)
    slug = models.SlugField(max_length=255, unique=True)

    def __str__(self):
        return self.name
```

Hình 5 Mô hình dữ liệu Category

**Trường dữ liệu:**

- sub\_category: Lưu danh mục con.
- is\_sub: Kiểm tra xem danh mục đó có phải là danh mục con không.
- name: Trường lưu tên danh mục.
- slug: Trường lưu văn bản thu gọn của tên danh mục để sử dụng trong URL.

**Phương thức:**

- self.name: Trả về tên danh mục.

#### 2.2.5 payment (thanh toán)

```
class Payment_VNPay(models.Model):
    order_id = models.IntegerField(default=0, null=True, blank=True)
    amount = models.FloatField(default=0.0, null=True, blank=True)
    order_desc = models.CharField(max_length=200, null=True, blank=True)
    vnp_TransactionNo = models.CharField(max_length=200, null=True, blank=True)
    vnp_ResponseCode = models.CharField(max_length=200, null=True, blank=True)

class PaymentForm(forms.Form):
    order_id = forms.CharField(max_length=250)
    order_type = forms.CharField(max_length=20)
    amount = forms.IntegerField()
    order_desc = forms.CharField(max_length=100)
    bank_code = forms.CharField(max_length=20, required=False)
    language = forms.CharField(max_length=2)
```

Hình 6 Mô hình dữ liệu Payment\_VNPay và biểu mẫu PaymentForm

### **Trường dữ liệu:**

- **Payment\_VNPay model:**
  - order\_id: Mã đơn hàng.
  - amount: Số tiền thanh toán.
  - order\_desc: Mô tả đơn hàng.
  - vnp\_TransactionNo: Mã giao dịch từ VNPAY.
  - vnp\_ResponseCode: Mã phản hồi từ VNPAY.
- **PaymentForm form:**
  - order\_id: Mã đơn hàng (dạng chuỗi).
  - order\_type: Loại đơn hàng (dạng chuỗi).
  - amount: Số tiền thanh toán (dạng số nguyên).
  - order\_desc: Mô tả đơn hàng (dạng chuỗi).
  - bank\_code: Mã ngân hàng (dạng chuỗi, không bắt buộc).
  - language: Ngôn ngữ giao dịch (dạng chuỗi).

## **2.3 Môi quan hệ giữa các đối tượng trong cơ sở dữ liệu**

Cơ sở dữ liệu của hệ thống được thiết kế theo mô hình quan hệ, trong đó các bảng được liên kết với nhau thông qua khóa chính – khóa ngoại. Dưới đây là mô tả chi tiết các mối quan hệ chính:

### **2.3.1 Category và Product:**

- **Mối quan hệ:** Quan hệ nhiều-nhiều (ManyToMany)
- **Giải thích:**
  - Mỗi Product có thể thuộc nhiều Category, và một Category có thể chứa nhiều Product.
  - Điều này được thể hiện qua trường category trong bảng Product với quan hệ ManyToManyField.

### **2.3.2 User và Order:**

- **Mối quan hệ:** Quan hệ một-nhiều (ForeignKey)
- **Giải thích:**
  - Một User có thể tạo nhiều Order, nhưng mỗi Order chỉ thuộc về một User.
  - Trường customer trong bảng Order là một khóa ngoại tham chiếu đến User.

### **2.3.3 Order và OrderItem:**

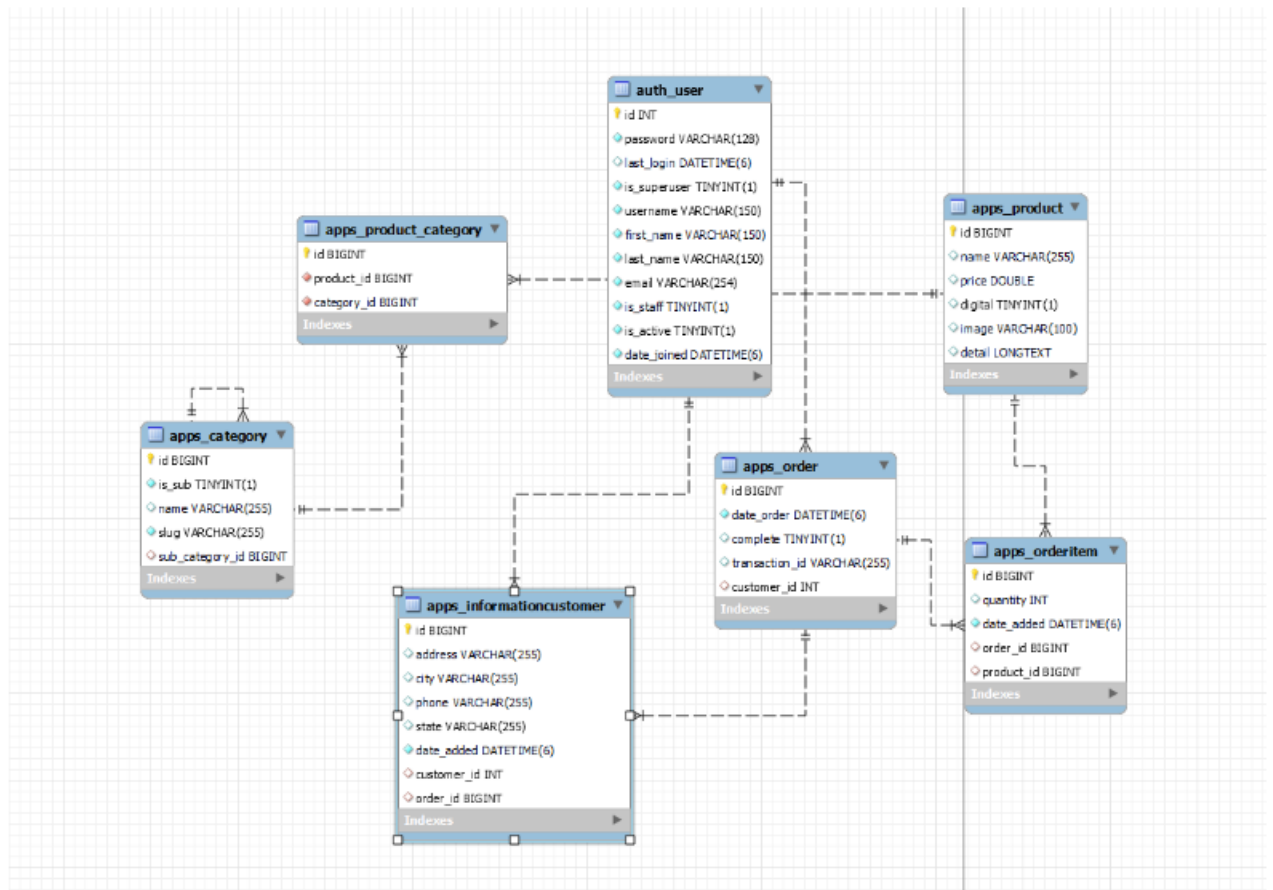
- **Mối quan hệ:** Quan hệ một-nhiều (ForeignKey)
- **Giải thích:**

- Một Order có thể có nhiều OrderItem, nhưng mỗi OrderItem chỉ thuộc về một Order.
- Trường order trong bảng OrderItem là một khóa ngoại tham chiếu đến Order.

### 2.3.4 Product và OrderItem:

- Mỗi quan hệ: Quan hệ một-nhiều (ForeignKey)
- Giải thích:
  - Một Product có thể xuất hiện nhiều lần trong các OrderItem khác nhau, nhưng mỗi OrderItem chỉ có một Product.
  - Trường product trong bảng OrderItem là một khóa ngoại tham chiếu đến Product.

## 2.4 Thiết kế cơ sở dữ liệu

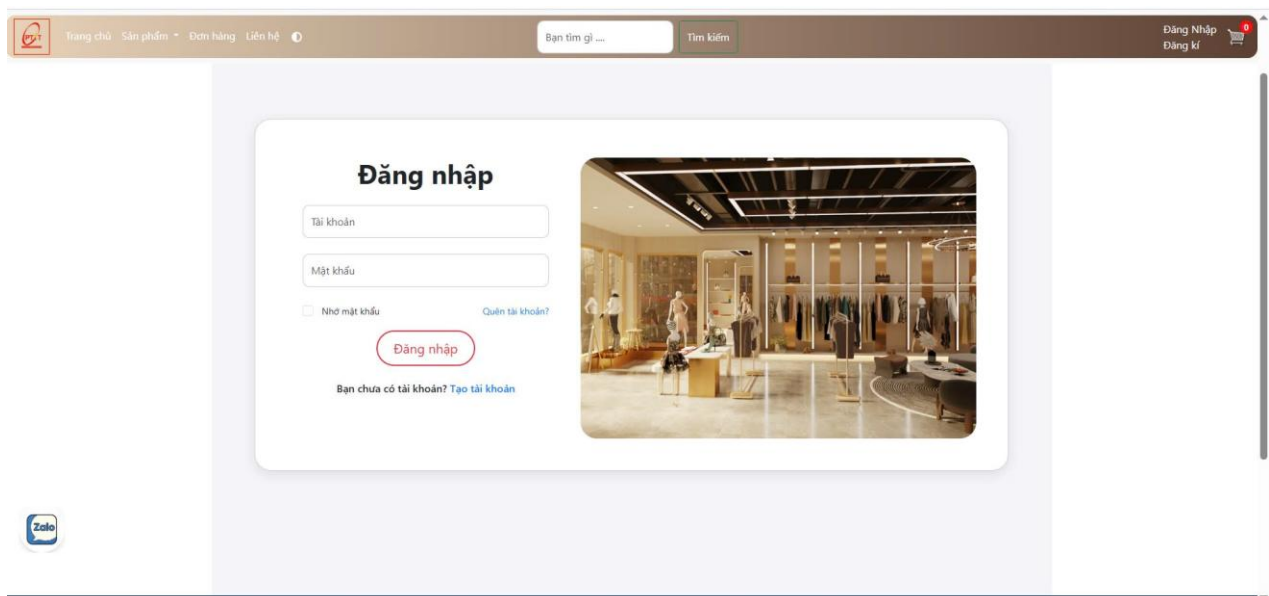


Hình 7 Thiết kế cơ sở dữ liệu

## 2.5 Thiết kế giao diện

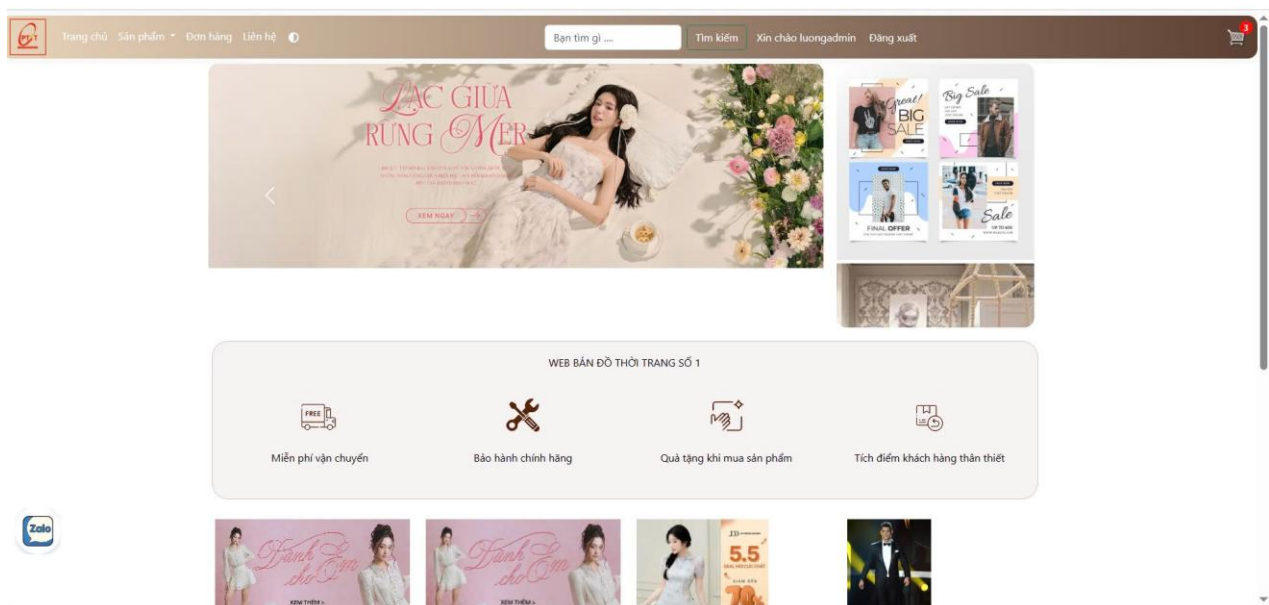
Các màn hình chính trong website:

- Giao diện đăng nhập người dùng trên website bán đồ thời trang:



Hình 8 Giao diện trang đăng nhập của website bán hàng

- Giao diện trang chủ của website bán đồ thời trang, hiển thị sau khi admin đăng nhập:



Hình 9 Giao diện trang chủ sau khi người dùng đăng nhập của website

- Giao diện đăng ký tài khoản trên website bán hàng thời trang:



Hình 10 Giao diện đăng ký tài khoản của website

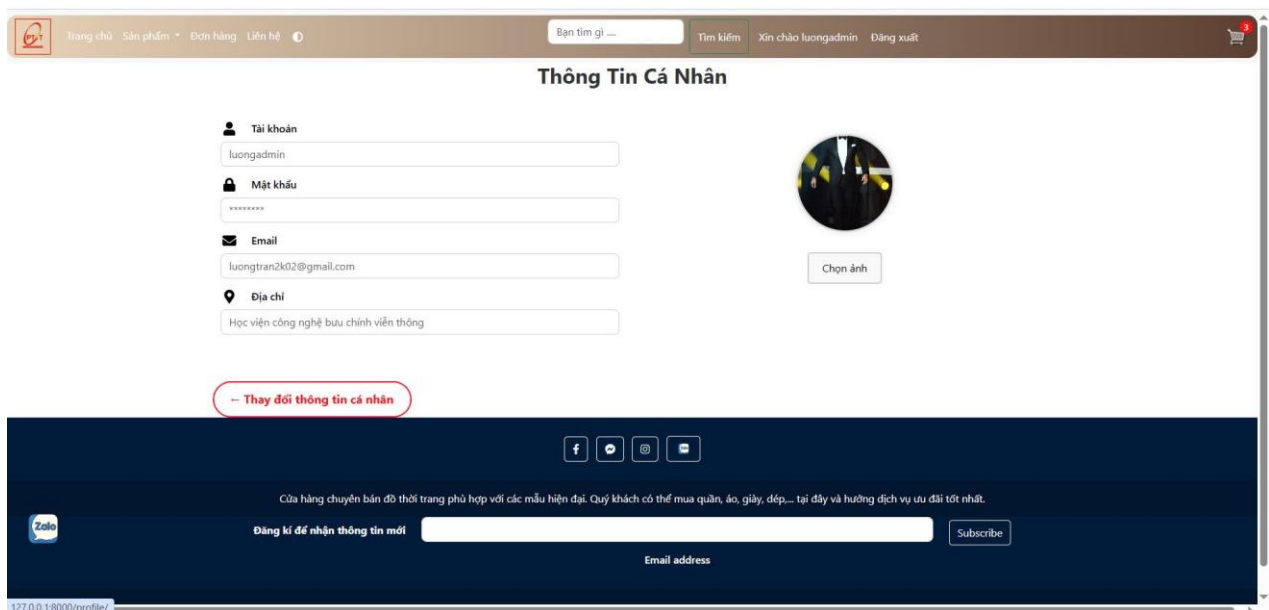
- Giao diện giỏ hàng (cart) của người dùng, hiển thị thông tin sản phẩm đã chọn, số lượng, giá và tổng tiền, cùng với các nút “Tiếp tục chọn sản phẩm” và “Thanh toán”:

Tên sản phẩm	Giá	Số lượng	Tổng
Test 1	1.0	2	2.0
Váy nữ đẹp	500000.0	1	500000.0

Hình 11 Giao diện giỏ hàng (cart) của người dùng

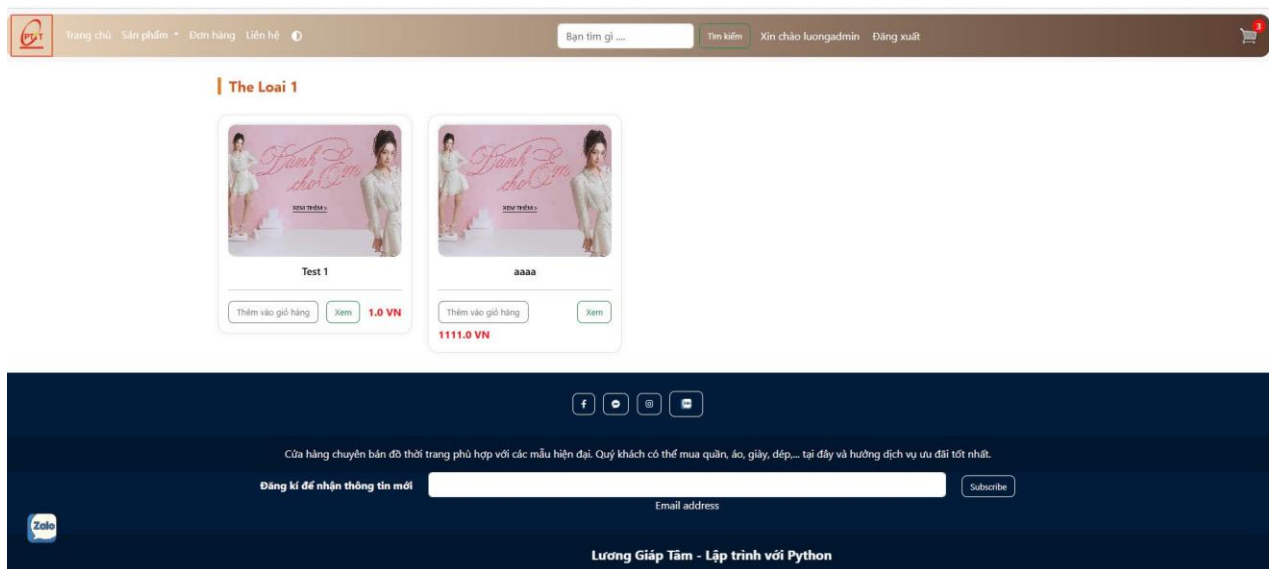
- Giao diện trang thông tin cá nhân (profile) của người dùng đã đăng nhập, với các trường:
  - tài khoản
  - mật khẩu
  - email
  - địa chỉ
  - ảnh đại diện





Hình 12 Giao diện trang thông tin cá nhân (profile) của người dùng đã đăng nhập

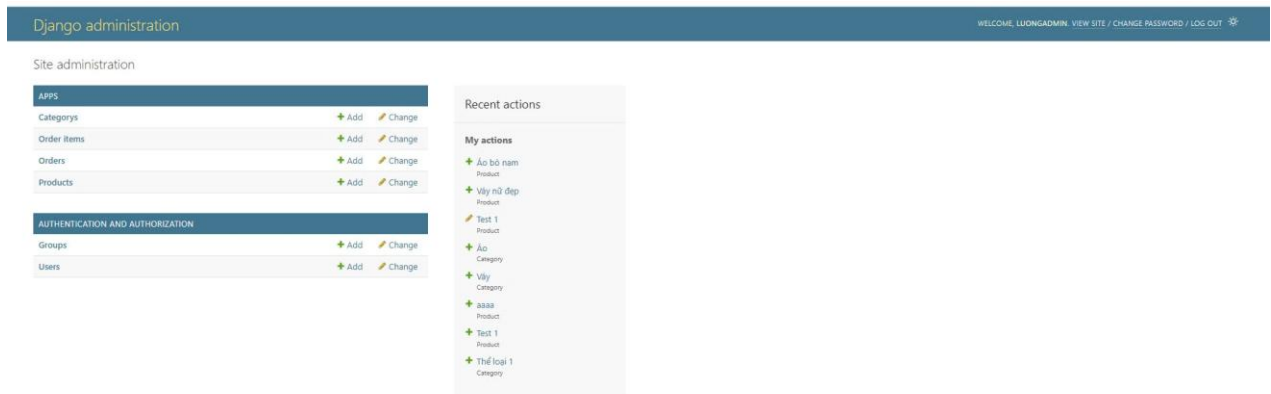
- Giao diện danh sách sản phẩm theo danh mục, trong đó hiển thị thông tin sản phẩm như tên, giá, nút xem chi tiết và thêm vào giỏ hàng.



Hình 13 Giao diện danh sách sản phẩm theo danh mục

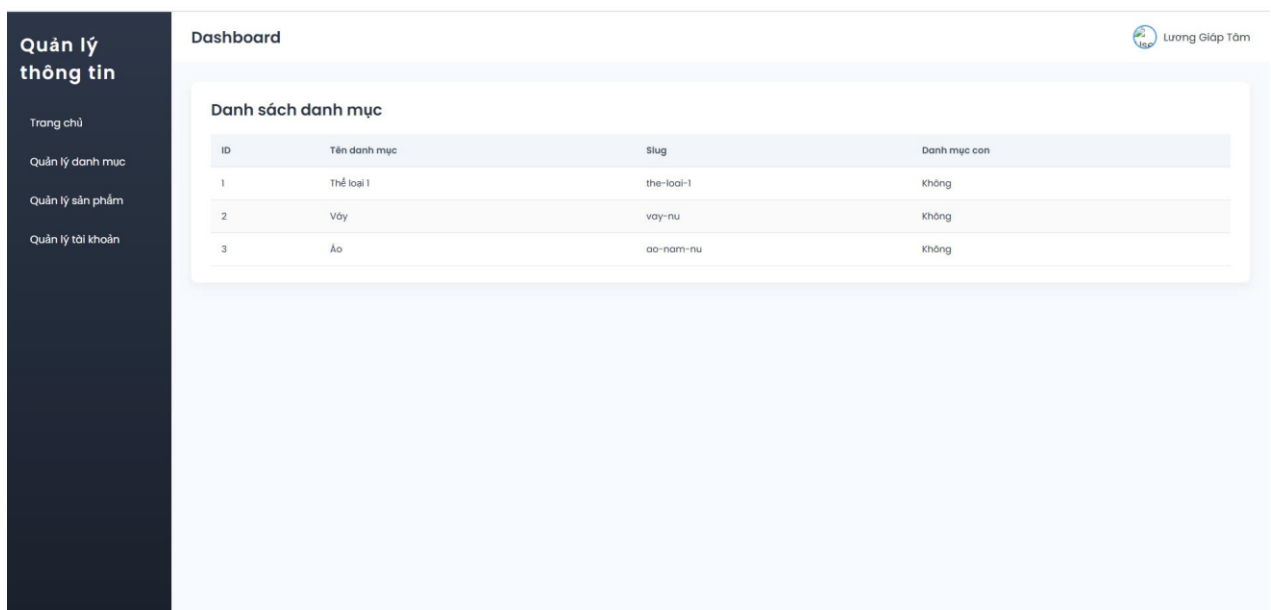
- Giao diện Django Admin – bảng điều khiển quản trị mặc định của Django, nơi quản lý các bảng dữ liệu như:
  - Categories
  - Order items
  - Orders
  - Products
  - Groups

- Users



Hình 14 Giao diện người quản trị

- Giao diện Dashboard – Quản lý danh mục sản phẩm: Giao diện hiển thị danh sách các danh mục sản phẩm trong hệ thống quản trị, bao gồm:
  - Tên danh mục
  - Slug
  - Thông tin danh mục con tương ứng.



Hình 15 Giao diện Dashboard – Quản lý danh mục sản phẩm

- Ngoài ra còn một số giao diện khác nữa.

## 2.6 Kết chương

Chương 2 tập trung phân tích hệ thống, làm rõ các yêu cầu chức năng, dữ liệu và mối quan hệ giữa các thực thể. Việc xây dựng mô hình dữ liệu rõ ràng, kết hợp với thiết kế giao diện thân thiện đã tạo nền tảng vững chắc cho quá trình phát triển và triển khai ứng dụng.

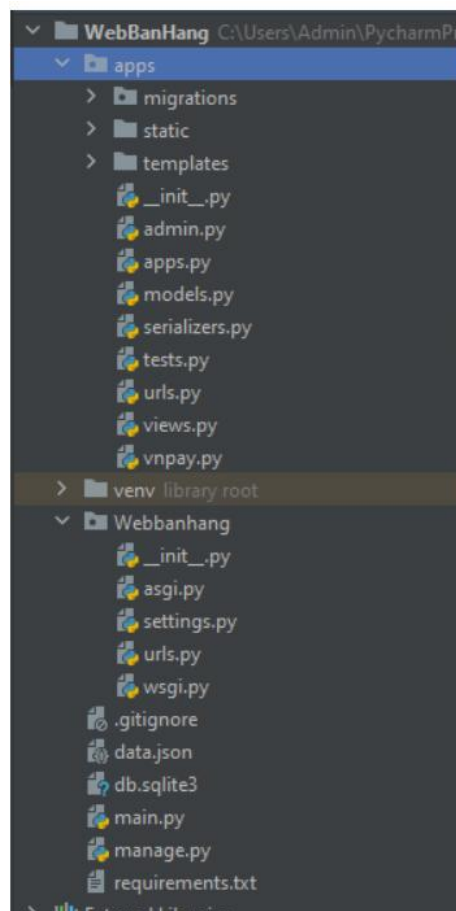
## CHƯƠNG 3. MÔ TẢ CÁCH HOẠT ĐỘNG CỦA HỆ THỐNG

### 3.1 Các thành phần chính

Hệ thống website thương mại điện tử được xây dựng bằng Django hoạt động theo mô hình client-server, trong đó Django đảm nhiệm vai trò server-side để xử lý yêu cầu, quản lý dữ liệu, và render giao diện HTML động gửi về cho trình duyệt người dùng. Cách thức hoạt động của hệ thống có thể chia thành ba phần chính:

- Người dùng (Client): Tương tác với website thông qua trình duyệt (trang chủ, tìm kiếm sản phẩm, đặt hàng, đăng nhập...).
- Máy chủ ứng dụng (Django Server):
  - URL Router: Tiếp nhận request, phân phối đến các view tương ứng.
  - Views (Xử lý logic): Nhận yêu cầu từ client, xử lý dữ liệu, gọi model và gửi dữ liệu đến template.
  - Models (ORM): Tương tác với cơ sở dữ liệu thông qua Django ORM.
  - Templates: Render HTML hiển thị cho người dùng.
- Cơ sở dữ liệu (SQLite): Lưu trữ thông tin người dùng, sản phẩm, đơn hàng, đánh giá, v.v.

### 3.2 Cấu trúc thư mục



Hình 16 Cấu trúc thư mục

Dưới đây là mô tả cấu trúc thư mục của Website:

- WebBanHang (Thư mục gốc trên cùng): Thư mục chính của dự án chứa toàn bộ các thành phần của ứng dụng
- Apps: Thư mục chứa ứng dụng con trong dự án
- Webbanhang: Thư mục này là phần cài đặt (settings) chính của dự án Django, bao gồm các tệp cấu hình như settings.py, urls.py.
- static: Thư mục chứa các tệp tĩnh (CSS, JavaScript, hình ảnh) phục vụ cho phân giao diện người dùng.
- templates: Thư mục chứa các tệp HTML, được sử dụng để render giao diện người dùng trong Django.
- venv: Thư mục của môi trường ảo Python, giúp quản lý các thư viện và phụ thuộc của dự án mà không ảnh hưởng đến hệ thống toàn cục.
- db.sqlite3: Tệp cơ sở dữ liệu SQLite của dự án, nơi lưu trữ dữ liệu của ứng dụng.
- manage.py: Tệp chính để quản lý dự án Django, giúp thực hiện các lệnh như chạy server, di chuyển cơ sở dữ liệu, tạo ứng dụng, v.v.

### **3.3 Quy trình hoạt động tổng quát**

#### ***3.3.1 Tổng quan hoạt động hệ thống***

Hệ thống thương mại điện tử được xây dựng nhằm cung cấp nền tảng cho người dùng mua sắm trực tuyến. Tổng quan hoạt động hệ thống có thể tóm gọn qua các bước sau:

1. Người dùng truy cập website thông qua trình duyệt, bắt đầu từ trang chủ (home).
2. Nếu chưa đăng nhập, người dùng có thể thực hiện:
  - Xem sản phẩm
  - Tìm kiếm sản phẩm
  - Xem danh mục
  - Đăng ký tài khoản hoặc đăng nhập để sử dụng các chức năng nâng cao.
3. Khi đăng nhập thành công, người dùng có thể:
  - Thêm sản phẩm vào giỏ hàng
  - Thực hiện thanh toán
  - Theo dõi đơn hàng, cập nhật thông tin cá nhân
4. Quản trị viên sử dụng giao diện admin của Django để:
  - Quản lý sản phẩm, đơn hàng, người dùng
  - Duyệt và xử lý giao dịch, cập nhật trạng thái thanh toán
5. Giao dịch thanh toán trực tuyến được xử lý thông qua VNPAY, bao gồm:

- Gửi yêu cầu thanh toán
- Nhận kết quả IPN (Instant Payment Notification)
- Phản hồi trạng thái thanh toán
- Lưu thông tin giao dịch

### 3.3.2 Tương tác nội bộ giữa các thành phần

Thành phần chính	Vai trò	Tương tác
<b>Model</b>	Đại diện cho cấu trúc dữ liệu (User, Product, Order, OrderItem, Payment)	Tương tác chặt với View để lưu/đọc dữ liệu
<b>View (views.py)</b>	Xử lý logic nghiệp vụ (xử lý đăng nhập, giỏ hàng, thanh toán, v.v.)	Giao tiếp với Model để thao tác DB, và trả dữ liệu qua context cho Template
<b>Template (HTML)</b>	Giao diện hiển thị tới người dùng	Nhận dữ liệu từ View và kết xuất HTML
<b>urls.py</b>	Định tuyến URL đến đúng View	Là cầu nối giữa frontend và backend
<b>Session và Authentication</b>	Quản lý đăng nhập, trạng thái người dùng	Sử dụng mặc định của Django để xử lý bảo mật
<b>VNPAY Module</b>	Xử lý thanh toán trực tuyến	Giao tiếp với API VNPAY để gửi và nhận kết quả giao dịch
<b>Admin site</b>	Giao diện quản lý cho admin	Tương tác trực tiếp với các model để thêm/sửa/xóa

*Bảng 1. Bảng mô tả tương tác nội bộ giữa các thành phần*

### 3.4 Kết chương

Chương 3 mô tả chi tiết quá trình hoạt động của hệ thống, từ truy cập, đăng nhập, thêm sản phẩm vào giỏ đến thanh toán và quản trị. Mô hình MVT của Django được áp dụng hiệu quả, đảm bảo tính phân tách giữa logic xử lý, dữ liệu và giao diện, từ đó nâng cao hiệu quả phát triển và bảo trì.

## KẾT LUẬN, KIẾN NGHỊ

### Kết luận

Sau quá trình nghiên cứu và triển khai, nhóm đã xây dựng thành công website bán hàng trực tuyến sử dụng Django framework. Hệ thống đáp ứng đầy đủ các chức năng cơ bản của một nền tảng thương mại điện tử như: quản lý sản phẩm, giỏ hàng, thanh toán trực tuyến (tích hợp VNPAY), đăng ký – đăng nhập, và giao diện quản trị.

Dự án không chỉ giúp nhóm áp dụng lý thuyết vào thực tế mà còn rèn luyện kỹ năng làm việc nhóm, tư duy logic và kỹ năng xử lý vấn đề trong phát triển phần mềm.

### Kiến nghị

- Trong tương lai, hệ thống có thể mở rộng thêm các tính năng như: gửi email xác nhận đơn hàng, thống kê theo thời gian thực hoặc tích hợp hệ thống AI.
- Đối với bài giảng, kiến nghị giảng viên có thể gợi ý thêm các hướng mở rộng để sinh viên phát triển sâu hơn dự án, giúp tăng tính thực tiễn và sáng tạo.
- Hệ thống nên được triển khai trên một hosting thực tế để kiểm thử với người dùng thật và đánh giá hiệu suất hoạt động.

## TÀI LIỆU THAM KHẢO

- [1] Django Project Official Documentation. <https://docs.djangoproject.com/>
- [2] VNPAY Developer Portal. <https://sandbox.vnpayment.vn/apis/docs>
- [3] Bootstrap Documentation. <https://getbootstrap.com/>
- [4] Mozilla Developer Network (MDN) Web Docs. <https://developer.mozilla.org/>
- [5] W3Schools – HTML, CSS, JS Tutorials. <https://www.w3schools.com/>
- [6] SQLite Official Documentation. <https://www.sqlite.org/docs.html>
- [7] Nguyễn Đức Hoàng. *Giáo trình Lập trình Web với Django*. NXB Đại học Quốc gia TP.HCM, 2021.